

**ESCUELA SUPERIOR**

**POLITÉCNICA DEL LITORAL**

Facultad de Ingeniería Eléctrica y Computación

**Diseño de Software**

**Proyecto Segundo Parcial**

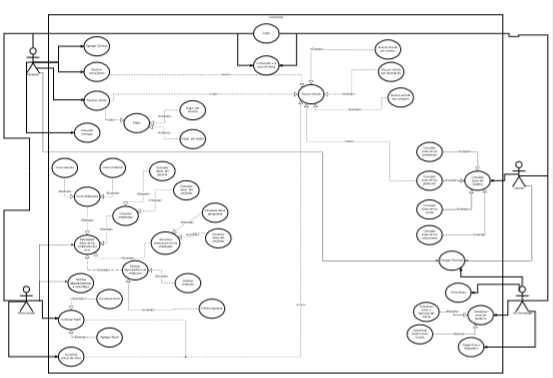
**Integrantes:**

* Luis Eduardo Salazar Castro
* Carlos Sánchez Acosta
* Diego Yance Álvarez

# Descripción del Sistema

Nuestra propuesta es diseñar un software capaz de suplir las necesidades de cualquier negocio que necesite de un sistema de ventas y manejo de inventario de productos que dicho negocio oferte, mediante la aplicación como enfoque principal del paradigma de programación orientada a objetos. Esto se debe a que el sistema requiere una representación de modelos que posean sus propios atributos y métodos, además de aplicar otros paradigmas como programación orientada a aspectos. Este paradigma es muy útil al momento de poder controlar y verificar la sesión de los usuarios conectados con el sistema ya que por ejemplo al momento de consultar o manipular la base de datos, esta información es muy delicada, entonces el poder controlar las credenciales de sesión es muy importante como mecanismo de seguridad. Para el manejo de la base de datos se usará la herramienta de MySQL debido a que es una base de datos relacional y trabaja muy bien con el paradigma orientada a objetos ya que los datos están relacionados entre sí. Se empleará la herramienta Docker, tanto para el programa como para la base de datos, con el propósito de que se pueda utilizar el sistema en cualquier computador sin ningún inconveniente.

# Diagrama Casos de uso



# Casos de uso

|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **CDU-A01** |
| **Titulo** | Controlar Stock |
| **Autor** | Diego Yance, Luis Salazar, Carlos Sánchez |
| **Actor principal** | Administrador |
| **Actor secundario** | NA |
| **Usa** | NA |
| **Extiende** | Extiende a Agregar Stock y a Actualizar Stock |
| **Descripción** | El administrados revisa el stock que recibió de la bodega y decide qué hacer con los productos que le llegaron y como introducirlos a la base de datos del sistema. |
| **Precondición** | El administrador debe haber ingresado en su cuenta del sistema |
| **Secuencia normal** | 1. El administrador recibe los productos 2. Interactúa con el sistema para abrir la ventana de controlar stock 3. Ingresa la información de los productos dependiendo de si el producto ya existe en la tienda o es nuevo. 4. El sistema guarda la información en la base de datos local y remota |
| **Secuencia alternativa** | 4.a La base de datos remota no está disponible   1. Se presenta el mensaje de desconexión al administrador 2. Se procede a guardar la operación en la base de datos local 3. Se procede a guardar la operación en un archivo log |
| **Postcondición** | El administrador logra ingresar de manera correcta los productos que recibió de la bodega |

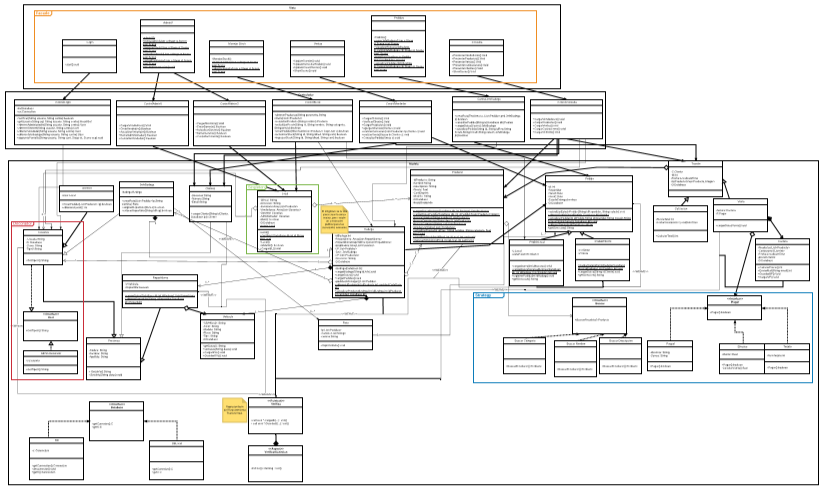
|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **CDU-A02** |
| **Titulo** | Actualizar Stock |
| **Autor** | Diego Yance, Luis Salazar, Carlos Sánchez |
| **Actor principal** | Administrador |
| **Actor secundario** | NA |
| **Usa** | NA |
| **Extiende** | Extiende de Controlar Stock |
| **Descripción** | El administrador modifica la cantidad de un producto ya existente en la base de datos para que refleje la cantidad total que se tiene después de la entrega de la bodega |
| **Precondición** | El administrador debe haber ingresado en el sistema, y debe estar realizando el ingreso del pedido de la bodega. |
| **Secuencia normal** | 1. El administrador encontró el producto en el sistema 2. Se procede a cambiar el valor de la cantidad disponible. 3. El sistema guarda el cambio en la base de datos remota y local |
| **Secuencia alternativa** | 4.a La base de datos remota no está disponible   1. Se presenta el mensaje de desconexión al administrador 2. Se procede a guardar la operación en la base de datos local 3. Se procede a guardar la operación en un archivo log |
| **Postcondición** | El administrador actualizo con éxito la cantidad del producto deseado |

|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **CDU-A03** |
| **Titulo** | Ingresar Stock |
| **Autor** | Diego Yance, Luis Salazar, Carlos Sánchez |
| **Actor principal** | Administrador |
| **Actor secundario** | NA |
| **Usa** | NA |
| **Extiende** | Extiende de Controlar Stock |
| **Descripción** | El administrador ingresa un nuevo producto al sistema y se lo guarda en la base de datos. |
| **Precondición** | El administrador debe haber ingresado en el sistema y debió acceder a esta sección. |
| **Secuencia normal** | 1. El administrador no encontró el producto en el sistema 2. Se le presenta la ventana para llenar todos los datos necesarios del producto 3. El administrador llena los campos 4. El sistema guarda los cambios en la base de datos remota y local |
| **Secuencia alternativa** | 4.a La base de datos remota no está disponible   * 1. Se presenta el mensaje de desconexión al administrador   2. Se procede a guardar la operación en la base de datos local   3. Se procede a guardar la operación en un archivo log |
| **Postcondición** | El administrador ingreso de manera exitosa el nuevo producto en el sistema y en las bases de datos. |

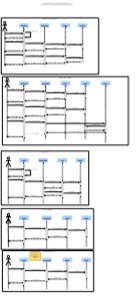
|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **CDU-A04** |
| **Titulo** | Actualizar Precio de Venta |
| **Autor** | Diego Yance, Luis Salazar, Carlos Sánchez |
| **Actor principal** | Administrador |
| **Actor secundario** | NA |
| **Usa** | NA |
| **Extiende** | NA |
| **Descripción** | El administrador desea cambiar el precio de un producto |
| **Precondición** | El administrador debe haber ingresado en el sistema |
| **Secuencia normal** | 1. El administrador entra al sistema para manejar productos 2. Se busca el producto usando el nombre 3. El sistema le presenta la información del producto 4. El administrador verifica que el producto este correcto y actualiza el precio. 5. El sistema guarda el cambio en la base de datos local y remota |
| **Secuencia alternativa** | 4.a La base de datos remota no está disponible   * 1. Se presenta el mensaje de desconexión al administrador   2. Se procede a guardar la operación en la base de datos local   3. Se procede a guardar la operación en un archivo log |
| **Postcondición** | El administrado actualizo exitosamente el precio del producto deseado. |

|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **CDU-A05** |
| **Titulo** | Solicitar Abastecimiento a la Bodega |
| **Autor** | Diego Yance, Luis Salazar, Carlos Sánchez |
| **Actor principal** | Administrador |
| **Actor secundario** | Jefe de Bodega |
| **Usa** | NA |
| **Extiende** | NA |
| **Descripción** | El administrador analizo el inventario del local y desea enviarle un pedido a la bodega para recibir más productos |
| **Precondición** | El administrador debe haber ingresado y debe saber que productos desea ordenar. |
| **Secuencia normal** | 1. El administrador entra al sistema para crear un pedido 2. El sistema le presenta las posibles bodegas de las que se puede ordenar 3. El administrador escoge una bodega 4. El sistema le presenta los productos disponibles 5. El administrador selecciona los ítems que desee y se guardan en una lista. 6. Se crea el pedido y se lo guarda en la base de datos remota y local. |
| **Secuencia alternativa** | 6.a No se pudo conectar a la base de datos remota   1. El sistema del local no se pudo conectar a la base de datos remota. 2. El pedido no pudo ser enviado 3. Se guarda la operación en un archivo log |
| **Postcondición** | El administrador solicita una lista de productos a la bodega con éxito. |

# Diagrama de Clases

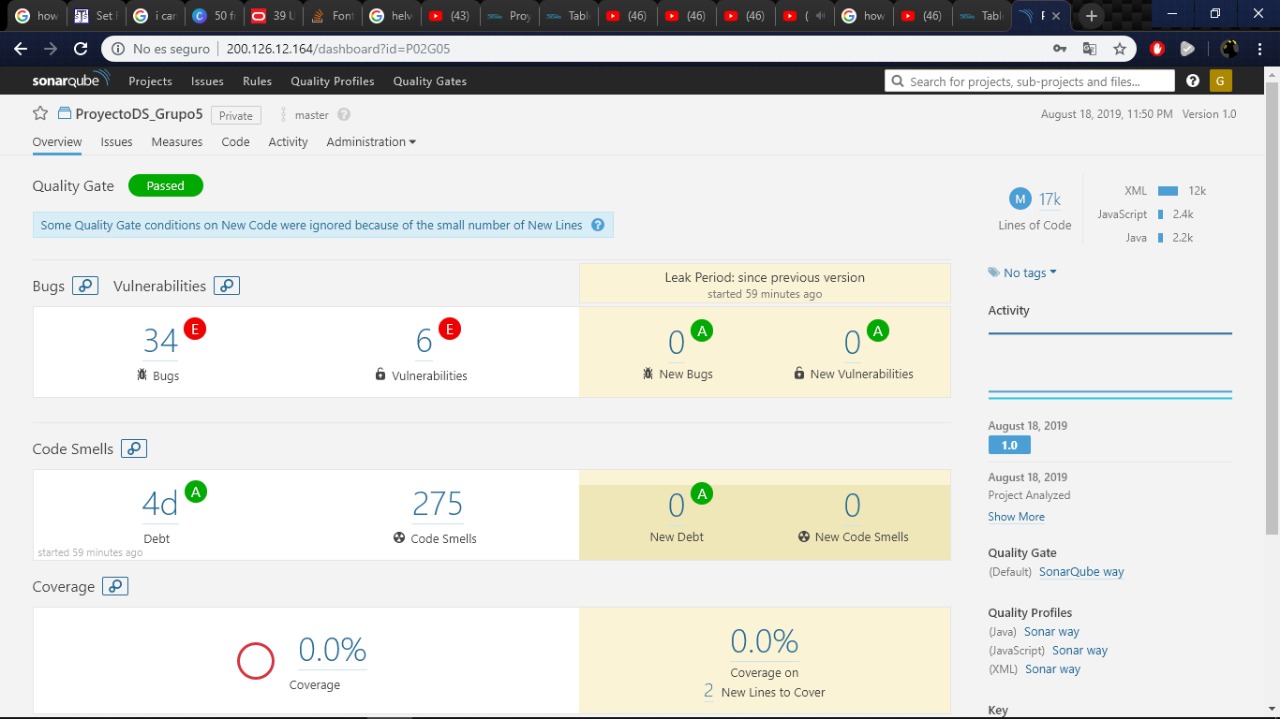


# Diagrama de Secuencias

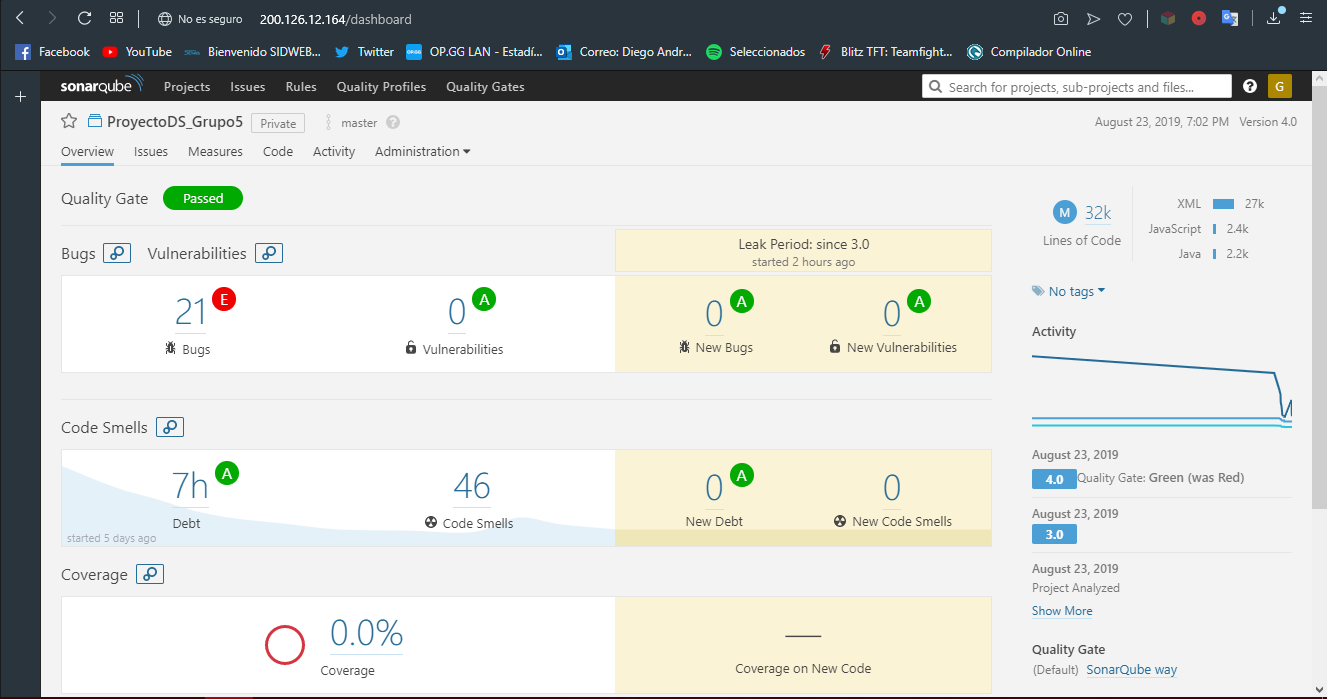


# SONARQUBE

**Antes**

****

**Después**

****

Docker

Para el sistema elaborado, se decidió utilizar un contenedor para almacenar la base de datos. A continuación, se detallan las ventajas de su uso:

* Rapidez en la ejecución de los Querys.
* Evitar el uso de un software externo inestable (específicamente el constante problema que tiene MySQL Workbench al cerrarse sin previo aviso.)

La única desventaja que presento el uso del contenedor fue la sensibilidad a las mayúsculas y minúsculas al referirse a distintas tablas, procedures, etc/.

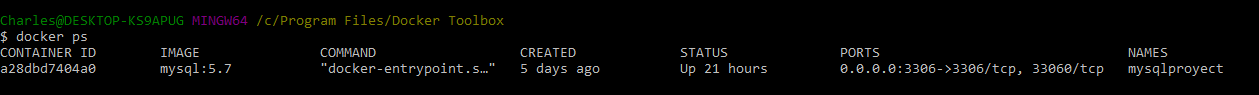
Se detalla cómo fue utilizado y los comandos requeridos:

Comando para crear el contenedor:



Se crea el contenedor con el nombre mysqlproyect, se establece el puerto a utilizar y otras banderas para ciertas especificaciones. Por ejemplo, la que indica el uso de la imagen mysql 5.7. Además, se indica que el usuario root tiene esa misma clave al acceder a la base de datos. También se le indica al contenedor trabajar con un volumen, esto reduce el riesgo de perder el contenido al compartir un archivo de trabajo entre el contenedor y la máquina local.

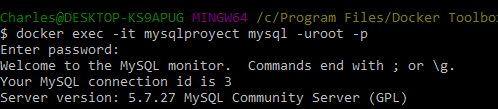
Inmediatamente el contenedor se levanta, es posible verificarlo con el uso del comando ps.



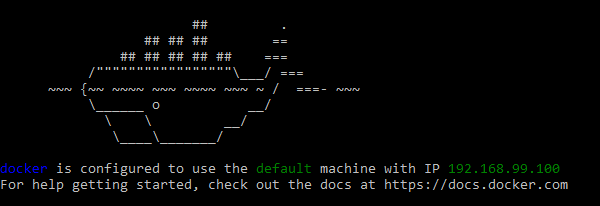
Indica como satisfactoriamente se encuentra corriendo.

Para empezar a trabajar en el contenedor se utiliza el comando exec., indicando que se desea trabajar en el entorno de mysql.

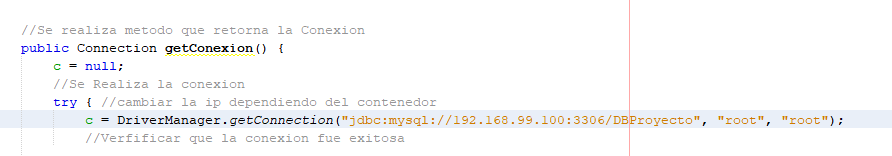
Se ingresa la clave configurada anterior y finalmente se procede a trabajar con la base de datos.

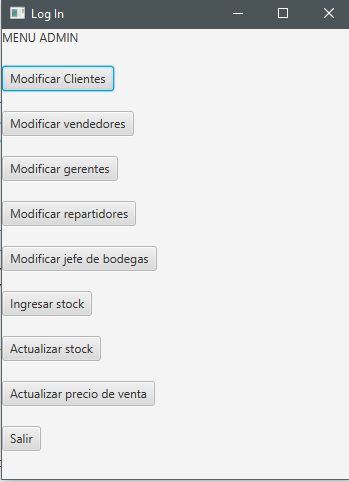


Para la conexión con el sistema desarrollado, se utiliza la ip del contenedor que tiene por defecto.



Se ingresa todos los datos de la configuración del contenedor en la clase BD.java





¡Funciona!

# Pruebas Unitarias

